

**In the Claims:**

The claims are as follows:

1. (Previously presented) A method for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement, the method being implemented by execution of program instructions by a computer, the program instruction being stored in computer-readable memory, the method comprising the steps of:

(a) analysing the callable procedure, the callable procedure comprising a branch condition under which control flow code directs program flow from the branch condition to a code branch of two or more code branches, each said code branch being within the callable procedure and branching from the branch condition to program code within the callable procedure, said analysing the procedure identifying the control flow code and the code branches;

(b) identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new procedures collectively comprise the two or more code branches;

(c) recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch;

(d) for the or each call statement, scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters directed to said control flow code by the call statement and modifying the call statement to replace the call to the original procedure by a call to the corresponding new procedure.

2. (Previously presented) The method of claim 1, wherein step (a) comprises constructing a control flow graph for the procedure, the control flow graph comprising a branching node representative of said control flow code and further nodes representing respective ones of said code branches.

3. (Previously presented) The method of claim 1, wherein one or more of said code branches each itself comprises two or more code branches and an item of control flow code for directing program flow to those code branches and wherein between said steps (a) and (b), said method comprising the further steps of:

(e) for each item of control flow code, before identifying any new procedure in accordance with step (b) of the method, checking for compliance between one or more predetermined rules for the software and the software should step (b) and following steps of the method take place; and

(f) for that item of control flow code, continuing with step (b) and the following steps of the method only in the event of such compliance.

4. (Previously presented) The method of claim 3, wherein step (e) comprises the application of a cost-analysis algorithm based on predetermined rules about the length of the software.

5. (Previously presented) The method of claim 1, said method including optimising the or each new procedure for which a call parameter is a constant by propagating that constant through the new procedure.

6. (Previously presented) The method of claim 1, said method including analysing a call statement, calling parameters and an associated new procedure to determine if they are compliant with predetermined in-lining rules and, if they are so compliant, replacing said call statement by a copy of the new procedure.

7. (Previously presented) A method for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement, the method being implemented by execution of program instructions by a computer, the program instructions being stored in a computer-readable memory, the method comprising the steps of:

(a) constructing a control flow graph for the callable procedure, the control flow graph comprising one or more branching nodes each representative of respective control flow code and, for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node, wherein said one or more branching nodes and said respective code branches are contained within the callable procedure;

(b) considering each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered;

(c) recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition;

(d) for each said call statement, scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement; and

(e) modifying the call statements to call said new procedures.

8. (Previously presented) A computer system comprising program instructions adapted to be executed for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement the program instructions being stored in a computer-readable memory, the program instructions comprising:

(i) analysing means for analysing the callable procedure, the callable procedure comprising a branch condition under which control flow code directs program flow from the branch condition to a code branch of two or more code branches, each said code branch being within the callable procedure and branching from the branch condition to program code within the callable procedure, said means for analysing the procedure comprising means for identifying the control flow code and the code branches;

(ii) identifying means for identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new procedures collectively comprise the two or more code branches;

(iii) recording means for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch;

(iv) scanning means operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters supplied by the call statement; and

(v) modifying means for modifying the call statement to call the corresponding new procedure.

9. (Previously presented) The computer system of claim 8, wherein said analysing means is operable for storing data representing the nodes and edges of a control flow graph for the procedure, said nodes including a branching node representative of said control flow code and further nodes representative of respective ones of said code branches.

10. (Previously presented) The computer system of claim 8, wherein one or more of said code branches each itself comprises two or more code branches and an item of control flow code for directing program flow to those code branches and wherein the system comprises checking means which is operable to check for compliance between one or more predetermined rules for the software and the software should said identifying means identify any new procedure.

11. (Previously presented) The computer system of claim 10, wherein said checking means is

operable for checking compliance with a cost-analysis algorithm based on predetermined rules about the length of the software.

12. (Previously presented) The computer system of claim 8, said program instructions including means for optimising the or each new procedure for which a call parameter is a constant by propagating that constant through the new procedure.

13. (Previously presented) The computer system of claim 8, said program instructions including means for analysing a call statement, calling parameters and an associated new procedure to determine if they are compliant with predetermined related to in-lining and, if they are so compliant, replacing said call statement by a copy of the new procedure.

14. (Previously presented) A computer system comprising program instructions adapted to be executed for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement, the program instructions being stored in a computer-readable memory, the program instructions comprising:

(i) means for storing data representing a control flow graph for the callable procedure, the control flow graph comprising one or more branching nodes each representative of respective control flow code and, for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node, wherein said branching nodes, said further nodes, and said code branches are contained within the callable procedure;

(ii) means for traversing the control flow graph to consider each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered;

(iii) means for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition;

(iv) means for scanning the entries in said list to determine for each call statement, an entry for which there is correspondence between said branch condition and call parameters supplied by the call statement; and

(v) means for modifying the call statements to call said new procedures.

15. (Previously presented) A computer program product comprising computer code stored in a computer-readable memory, the computer code when executed by a computer implements a method

for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement said computer code comprising:

(i) a first computer code portion for analysing the callable procedure, the callable procedure comprising a branch condition under which control flow code directs program flow

from the branch condition to a code branch of two or more code branches, each said code branch being within the callable procedure and branching from the branch condition to program code within the callable procedure, said analysing the procedure identifying the control flow code and the code branches;

(ii) a second computer code portion for identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new procedures collectively comprise the two or more code branches;

(iii) a third computer code portion for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch;

(iv) a fourth computer code portion operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters supplied by the call statement; and

(v) a fifth computer code portion for modifying the call statement to call the corresponding new procedure.

16. (Canceled)

17. (Previously presented) A computer program product for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call



statement and which has two or more code branches and control flow code for directing program flow to the code branches, said program product comprising computer code which includes:

(i) a first computer code portion for analysing the procedure to identify said control flow code and said code branches;

(ii) a second computer code portion for identifying for each said code branch a new procedure containing the respective code branch;

(iii) a third computer code portion for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the branch conditions under which said control flow code directs program flow to the associated code branch;

(iv) a fourth computer code portion operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch conditions and call parameters supplied by the call statement; and

(v) a fifth computer code portion for modifying the call statement to call the corresponding new procedure,

said computer software implemented in the form of a body of computer code made available for downloading from a computer connected to a computer network.

18. (Previously presented) A computer program product comprising computer code stored in a computer-readable memory, the computer code when executed by a computer implements a method for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement, said computer code comprising:

(i) a first code portion for storing data representing a control flow graph for the callable procedure, the control flow graph comprising one or more branching nodes each representative of respective control flow code and, for each branching node, two or more further nodes respective code branches to which program flow is directed from the branching node, wherein said branching nodes, said further nodes, and said code branches are contained within the callable procedure;

(ii) a second code portion for traversing the control flow graph to consider each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered;

(iii) a third code portion for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition;

(iv) a fourth code portion operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement; and

(v) a fifth code portion for modifying the call statements to call said new procedures.

19. (Canceled)

20. (Previously presented) A computer program product for optimising computer software that includes one or more call statements and a procedure which is callable by the or each call statement, said program product comprising computer code which includes:

(i) a first code portion for storing data representing a control flow graph for the procedure, the control flow graph comprising one or more branching nodes each representative of respective control flow code and, for each branching node, two or more further nodes representing respective code branches to which program flow is directed by the branching node;

(ii) a second code portion for traversing the control flow graph to consider each node in turn and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters and global variables, identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly from the node being considered;

(iii) a third code portion for recording a list of data entries corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition;

(iv) a fourth code portion operable, for each said call statement, for scanning the entries in said list to determine one for which there is correspondence between said branch condition and

call parameters supplied by the call statement; and

(v) a fifth code portion for modifying the call statements to call said new procedures, said computer software implemented in the form of a body of computer code made available for downloading from a computer connected to a computer network.